

PENCARIAN DATA PERPUTAKAAN DENGAN TOOL PROTEGE 3.4

R. Rhoedy Setiawan¹, Mukhamad Nurkamid²

^{1,2}Fakultas Teknik, Universitas Muria Kudus, Kudus 59127

e-mail : fl.umk@gmail.com

Abstrak

Makin banyak informasi yang ditambahkan, makin besar ukuran web dan semakin sulit pula untuk mencari informasi yang benar-benar yang diinginkan. Sumber data yang terbesar dan sangat berharga untuk setiap pengguna karena didalam web kumpulan dokumen-dokumen saling terhubung dan dapat diakses melalui koneksi internet. Setiap individu didunia bebas menambahkan konten dalam web-nya. Tujuan perpustakaan, pengarsipan, yang ada di institusi adalah untuk memelihara dokumen yang memiliki keilmuan yang dapat diakses dimasa depan sama mudahnya diakses pada saat ini. Dengan semakin kecilnya media data, penyimpanan menggunakan media digital menjadi sangat menarik. Keuntungan penyimpanan menggunakan media digital adalah pencarian dapat dilakukan tidak hanya di antara masukan katalog elektronik, tetapi dapat menampilkan sepanjang keseluruhan isi dokumen. Penciri dan pengembangan web mengesahkan tambahan informasi bagi konten web yang sering distilahkan dengan metadata. Istilah metadata ini sering dipakai dalam semantic web, yang menggambarkan pendekatan dalam menangani dan menyimpan dokumen. Saat ini dokumen mulai disajikan dalam format XML (Extensible Markup Language). Bahasa lain yang digunakan dalam mendukung visi Semantic Web selain XML adalah RDF/OWL. Selanjutnya RDF/OWL dengan kemampuan dan fasilitas yang dimilikinya mampu digunakan untuk merepresentasikan makna istilah dalam web hingga dapat diproses dalam mesin sehingga dapat dikelola dan dicari.

Kata Kunci : Search document, Approach method, Management process

1. PENDAHULUAN

Teknologi yang tepat untuk mengembangkan, membangun dan mengelola koleksi dokumen perpustakaan yang benar-benar diinginkan pengguna merupakan tujuan kedepan semua pihak. Tujuan perpustakaan, pengarsipan, yang ada di institusi adalah untuk memelihara dokumen yang memiliki harga keilmuan yang dapat diakses dimasa depan sama mudahnya diakses pada saat ini. Dengan semakin kecilnya media data, penyimpanan menggunakan media digital menjadi sangat menarik. Keuntungan penyimpanan menggunakan media digital adalah pencarian dapat dilakukan tidak hanya di antara masukan katalog elektronik, tetapi dapat menampilkan sepanjang keseluruhan isi dokumen. Permasalahan baru muncul disebabkan hard disk dan CD cepat ketinggalan jaman. Perkembangan sistem komputer dan komponennya lebih cepat dari pada perkembangan media digital, sehingga menyebabkan masalah kompatibilitas(keserjangan percepatan hardware yang tidak diikuti percepatan software). Penelitian ini adanya pengembangan web yang mengusulkan tambahan informasi bagi konten web yang sering distilahkan dengan metadata. Implementasi dengan pemanfaatan teknologi semantic web yang penulis lakukan yaitu pada perpustakaan dengan menampilkan bibliografi, baik untuk majalah, buku, jurnal, dan tesis.

2. TINJAUAN PUSTAKA

Perkembangan teknologi semantic web dapat diterapkan di segala macam bidang, misalnya seperti di dunia pendidikan, jurnalisme, transportasi dan inventory.

Penelitian mengenai semantic web beserta standar pendukungnya telah banyak dilakukan. Semantic web merupakan perluasan dari web yang ada pada saat ini, di mana informasi memiliki arti yang terdefinisi lebih baik, sehingga memungkinkan manusia dan komputer untuk bekerja sama lebih optimal, terutama dalam pengelolaan dan penyajian informasi (Lee, dkk, 2001).

Penelitian tentang standar yang berfokus pada content untuk mendukung semantic web telah banyak dilakukan. Beberapa penelitian-penelitian mahasiswa ilmu komputer Universitas Gadjah Mada sebelumnya yang dijadikan referensi adalah "Model Ontologi untuk Jadwal Penerbangan menggunakan Protege", (Minurita, 2005), "Pengembangan Model Ontologi untuk manajemen pengetahuan Staff Perusahaan", (Riyanti, 2006), "Penerapan Ontology Web Language pada pengarsipan berita Online", (Soraya, 2006) dan "Penerapan Teknologi Semantic Web pada Sistem Manajemen Perpustakaan Digital", (Karsanti, 2006).

Persamaan dari penelitian sebelumnya adalah menggunakan teknologi semantic web untuk membangun sistem aplikasi pencarian sedangkan perbedaannya hanya terletak pada subyek (domain) dan metode sistem.

3. METODE PENELITIAN

Dalam penelitian ini penulis menggunakan metode pemodelan Spiral (Pressman,1982), diantaranya meliputi:

a. Studi Literatur

Studi literatur mengenai *semantic web*, RDF/OWL dan situs-situs web yang relevan, studi kasus pencarian pada beberapa situs-situs edukasi melalui *digital library* dan mempelajari jenis-jenis properti untuk selanjutnya menjadi pertimbangan dalam pengembangan ontologi bibliografi pada sistem perpustakaan.

b. Perencanaan (Planning)

Menggambarakan mengenai sumber daya (*resource*), alokasi waktu (*timeline*) yang diperlukan untuk menyelesaikan pembuatan sistem dan informasi-informasi lainnya terkait dengan pengembangan sistem

c. Risk Analysis

Pada bagian ini merupakan analisa terhadap kekurangan-kekurangan yang terjadi pada sistem dengan tujuan untuk memperoleh masukan dan perbaikan sistem kedepan

d. Implementasi (Engineering)

Implementasi model ontologi dibangun dengan tool *protege 3.4*, kemudian aplikasi pencariannya diimplementasikan menggunakan *Java Server Pages* sebagai antar muka, *Jena API 2.5.6* untuk menghubungkan bahasa *Java* dengan basis data *RDF/OWL* dan *SPARQL* sebagai bahasa query *RDF/OWL*.

e. Construction and Release

Pengujian dilakukan untuk validasi apakah sistem yang dibangun sudah berjalan sesuai yang diharapkan dan memberikan hasil yang relevan.

4. HASIL DAN PEMBAHASAN

Perancangan ontologi untuk koleksi perpustakaan merupakan langkah awal dalam menentukan konsep dan domain yang akan dibangun. Noy dan McGuinness (2000) telah menjelaskan ada beberapa langkah-langkah yang harus diperhatikan salah satunya dengan menentukan konsep dan domain.

4.1. Penentuan Konsep dan Domain

Domain dari penelitian ini adalah koleksi data atau dokumen perpustakaan, yang pada dasarnya merepresentasikan koleksi semua dokumen yang dilengkapi dengan informasi dan disusun berdasar suatu klasifikasi dan dikelompokkan kedalam jenis-jenis yang sama (*Class*). Dokumen tersebut meliputi *article*, *proceedings*, *textbook*, *periodicreport* (*journal*, *magazine*, *newspaper*) dan *finalproject* (*bachelorthesis*, *masterthesis*, *phdthesis*).

4.2. Penentuan Daftar Terminologi

Sebelum menentukan daftar terminologi yang digunakan dalam ontologi, terlebih dahulu dilihat kemungkinan penggunaan ontologi yang sudah ada. Dengan pemanfaatan ontologi yang sudah ada waktu pengembangan software dapat cepat diselesaikan tanpa harus memulai dari awal. Untuk menggunakan ontologi yang sudah ada dapat diimport kedalam ontologi yang sedang dikembangkan. Dalam pembuatan aplikasi bibliografi pada sistem perpustakaan ini, diasumsikan belum terdapat ontologi yang dapat digunakan untuk pengembangan sehingga dalam hal ini seluruh komponen dari ontologi dikembangkan dari awal. Adapun terminologi yang penting dalam pengembangan ontologi aplikasi bibliografi ini sebagian besar mengadopsi elemen standar beberapa kosakata atau *term* dari *Dublin Core (dc)*, sebagian lagi didefinisikan sendiri dengan mempertimbangan penggunaan kosakata yang sudah dikenal oleh khalayak umum.

Penentuan daftar semua terminologi perlu diperhatikan karena dengan terminologi ini akan dipergunakan untuk membuat statemen sekaligus memberikan jawaban dari statemen yang dibuat sebelumnya. Sebagai contoh, terminologi yang terkait dengan konsep bibliografi meliputi: *publications*, *textbook*, *finalproject*, *article*, *journal*, *proceedings*, *author*, *publisher*, *address*, *referTo*, *publishedIn* dan *edition*.

4.3. Definisi Kelas dan Hirarki Kelas

Representasi dari hirarki kelas adalah mengelompokkan kelas-kelas dengan karakteristik yang sama. Suatu kelas dapat diturunkan dari kelas yang lain, dimana atribut dari kelas semula dapat diwariskan ke kelas yang baru. Pada aplikasi bibliografi ini dikembangkan dengan menggunakan pendekatan metode *top-down*, sebagai mana yang diungkapkan oleh *Uchold dan Gruninger (1996)* dalam makalah "*Ontologies: Principles, Methods and Applications*" bahwa ada beberapa pendekatan metode yang dapat digunakan untuk membangun struktur hirarki kelas diantaranya yaitu metode *top-down*, *bottom-up*, *combination*. Dengan metode *top-down*, kelas-kelas didefinisikan dari mulai konsep yang paling umum sampai konsep yang lebih spesifik.

Barclay dan Savage (2004) menyatakan bahwa karakteristik dari pendekatan sistem berorientasi obyek pada dasarnya kumpulan dari obyek yang saling berinteraksi. Kumpulan obyek-obyek yang sama ini dinamakan sebagai Kelas (*Class*). Didalam ontologi bibliografi pada perpustakaan didefinisikan dua kelas utama yaitu kelas *Person* dan kelas *Publications* yang kemudian dapat diperinci lagi sesuai dengan kebutuhan.

1. *Person*, kelas ini merepresentasikan informasi tentang pencipta dari suatu bibliografi. Kelas *Person* dapat dibagi kedalam subkelas yang lebih spesifik dengan nama kelas *Lecturer* dan kelas *Student*.

2. *Publications*, kelas ini merepresentasikan jenis-jenis referensi yang ada pada umumnya. Kelas ini adalah *superkelas* dari kelas *Article*, *Proceedings*, *PeriodicReport*, *TextBook* dan kelas *FinalProject*. Didalam kelas *PeriodicReport* dapat diuraikan lebih rinci kedalam subkelas *Journal*, *Magazine* dan *Newspaper*. Sementara untuk kelas *FinalProject* dapat diperinci menjadi beberapa subkelas dengan nama kelas *BachelorThesis*, kelas *MastersThesis* dan kelas *PhdThesis*.

4.4. Relasi Diagram Kelas (Class Diagram)

Suatu relasi adalah hubungan yang saling berkaitan antar kelas. Relasi mengijinkan sebuah kelas mengetahui atribut, operasi dan hubungan dengan kelas lainnya. Ada beberapa relasi didalam pendekatan desain berorientasi obyek salah satunya adalah relasi generalisasi. Generalisasi digunakan untuk memperlihatkan relasi pewarisan sifat antar kelas, sehingga pewarisan memperbolehkan suatu kelas untuk mewarisi semua atau sebagian atribut, operasi, relasi dan properti kelas lain. Diagram kelas atau *class diagram* menunjukkan interaksi antar kelas didalam sistem, sehingga dengan diagram kelas penyajian informasi yang dimiliki oleh setiap kelas dapat terlihat jelas. Kelas didefinisikan sebagai sebuah kumpulan dari obyek-obyek yang memiliki karakteristik yang sama dan menggambarkan perilaku dan atribut untuk semua obyek dari kelas (Garido, 2003).

4.5. Definisi Properti Kelas (Slot)

Setelah kelas diciptakan langkah selanjutnya adalah mendefinisikan properti kelas. Sebuah kelas jika berdiri sendiri tidak akan memberikan informasi yang cukup tanpa adanya properti yang melekat didalamnya. Dengan properti inilah sebuah kelas akan mempunyai nilai tambah dalam hal ini adalah informasi. Menurut Bernet (2001) bahwa properti dapat didefinisikan sebagai elemen-elemen data yang dimiliki oleh setiap kelas.

Untuk kelas *Lecturer* misalnya, properti yang dimiliki seperti *employeeId*, *email* dan *fullName*. Adapun properti-properti yang dimiliki kelas yang lainnya dapat disajikan seperti pada tabel 1 dan tabel 2.

Tabel 1: Properti Kelas *Lecturer*

Nama	Tipe	Kardinalitas	Keterangan
<i>employeeId</i>	<i>Int</i>	<i>Single</i>	Identitas dosen
<i>email</i>	<i>String</i>	<i>Multiple</i>	Email dosen
<i>fullName</i>	<i>String</i>	<i>Single</i>	Nama lengkap dosen

Kelas *Student* digunakan untuk menyimpan data-data mahasiswa. Tabel 2 menunjukkan beberapa properti yang dimiliki oleh kelas *Student* diantaranya, *studentNumber*, *year*, *email* dan *fullName*.

Tabel 2: Properti Kelas *Student*

Nama	Tipe	Kardinalitas	Keterangan
<i>studentNumber</i>	<i>Int</i>	<i>Single</i>	Identitas mahasiswa
<i>year</i>	<i>Int</i>	<i>Single</i>	Tahun masuk mahasiswa
<i>email</i>	<i>String</i>	<i>Multiple</i>	Email mahasiswa
<i>fullName</i>	<i>String</i>	<i>Single</i>	Nama lengkap mahasiswa

4.6. Gambaran Umum Aplikasi

Aplikasi yang dibangun dalam penelitian ini adalah aplikasi pencarian untuk koleksi perpustakaan yang merupakan salah satu bagian dari layanan sistem informasi perpustakaan dengan menggunakan pendekatan *thesaurus*. Metode *thesaurus* adalah sebuah skema klasifikasi untuk istilah-istilah yang saling berkait membentuk struktur bahasa sehingga sebuah kata dapat dipahami dengan kata lainnya (Pendis, 2008). Perancangan aplikasi pencarian berbasis teknologi *semantic web* ini menggunakan pendekatan ontologi, dimana didalam ontologi tersebut digunakan sebagai penyimpanan dokumen bibliografi. Selanjutnya untuk menemukan informasi yang terdapat didalam ontologi dapat digunakan SPARQL sebagai bahasa *query* RDF OWL. Jena API berfungsi sebagai *framework* yang menghubungkan antara ontologi dengan bahasa Java.

4.7. Diagram Use Case Pencarian Bibliografi

Use Case adalah deskripsi fungsi dari sebuah sistem yang mendefinisikan apa yang dilakukan oleh sistem dan elemen-elemennya. Bahwa pengguna dimungkinkan dapat memilih beberapa menu pencarian yang ada, seperti *keyword searching*, *simple searching* dan *advance searching*. Notasi "*extend*" memberikan alternatif apabila dengan *keyword searching* tidak memberikan hasil sehingga pengguna dapat memilih menu yang lain.

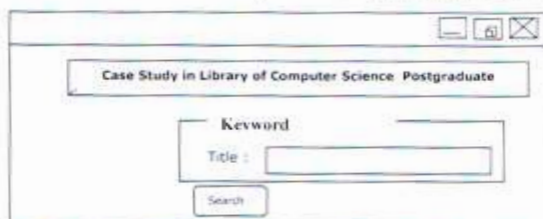
4.8. Aktivitas *Advanced Searching*

Pencarian selanjutnya adalah *advanced searching*, yang menyediakan lebih banyak kriteria pencarian yang dapat dipilih oleh pengguna, yaitu memilih kriteria berdasarkan penulis (*author*), penerbit (*publisher*), tahun (*year*) dan pembimbing (*advisor*).

Pencarian *advance searching* pada prinsipnya bertujuan menyediakan solusi dari kekurangan-kekurangan pencarian sebelumnya yang terdapat di *keyword searching* dan *simple searching*.

4.9. Perancangan Antarmuka Pengguna (GUI)

Antarmuka *user* atau lebih dikenal dengan GUI (*Graphical User Interface*) adalah bagian komponen dari sistem yang tidak dapat dipisahkan dengan *user*, karena dari GUI inilah *user* dapat berinteraksi dengan sistem dengan beberapa layanan yang telah disediakan. Rancangan antar muka bibliografi ini *user* dikategorikan menjadi tiga bagian, yaitu antarmuka untuk *keyword searching*, *simple searching* dan *advanced searching*. Antar muka *keyword searching* adalah antarmuka yang akan ditampilkan ketika pengguna menjalankan aplikasi dan terlihat di *web browser*. Di halaman *web* inilah pengguna dapat melakukan pencarian melalui beberapa menu yang telah disediakan oleh sistem. Antarmuka *keyword searching* menawarkan pencarian berdasarkan judul (*title*).



Gambar 1. Rancangan Antarmuka *Keyword Searching* Aplikasi Bibliografi

Adapun selain *keyword searching*, sistem akan menawarkan menu yang lain yang lebih spesifik seperti menu *simple searching* yang terlihat pada gambar 2. Pada menu ini akan diberikan tambahan kategori untuk memasukkan kata kunci berdasarkan nama penulis (*author*). Sehingga informasi yang diberikan kepada pengguna lebih lengkap. Implementasi antarmuka *keyword searching* dapat disajikan dalam gambar 2.



Gambar 2. Implementasi Antarmuka *Keyword Searching*

Pada fasilitas *keyword searching*, pengguna dapat memasukkan sembarang kata kunci (*keyword*) pada *textfield* judul yang tersedia. Kemudian hasil yang diinputkan akan dicocokkan dengan database RDF OWL dari rancangan ontologi yang telah didefinisikan. Hasil dari kata kunci yang diberikan dapat berupa semua informasi bibliografi, seperti jurnal, majalah, surat kabar, prosiding, tesis, skripsi, buku. Hasil pencarian judul dengan kata kunci "semantic web" pada *keyword searching* dapat disajikan dalam gambar 3.



Gambar 3. Hasil Sistem Pola untuk kata kunci "semantic web" pada Keyword Searching

5. KESIMPULAN

1. Dengan menggunakan SKOS (*Simple Knowledge Organization System*) sebagai pilar manajemen kosakata, membuktikan mesin dalam hal ini komputer mampu mengetahui relasi antar dokumen-dokumen yang terhubung (*machine-understandable*) sehingga dokumen tidak lagi berdiri sendiri.
2. Pada dasarnya ide *semantic web* bukanlah hanya berbicara mengenai pertukaran data, akan tetapi perlu di lihat bagaimana data didistribusikan dan disatukan, SKOS adalah salah satu mekanisme data dapat didistribusikan yang didukung bahasa RDF sebagai suatu format bahasa untuk pendistribusian data dilingkungan web.

DAFTAR PUSTAKA

- Baeza Y. R. dan Ribeiro N. B., 1999, *Modern Information Retrieval*, ACM Press, New York.
- Isaac, A. dan Summers, E., 2009, *SKOS Simple Knowledge Organization System Primer*, Recommendation, World Wide Web Consortium (W3C):
<http://www.w3.org/TR/2009/WD-skos-primer-20090615/>; available:
<http://www.w3.org/TR/skos-primer>
- Karsanti, H. T., 2006, *Penerapan Teknologi Semantic Web pada Sistem Manajemen Perpustakaan Digital*, Skripsi, Program Studi Ilmu Komputer FMIPA UGM, Yogyakarta.
- Miles, A., Matthews, B., Beckett, D., Brickley, D., Wilson, M dan Rogers, N., 2005, *SKOS - A Language to describe simple knowledge structures for the web*, Recommendation, World Wide Web Consortium (W3C):
<http://www.w3.org/TR/swbp-skos:a language to describe simple knowledge structures for the web>
- Miles, A. dan Bechhofer, S., 2009, *SKOS Simple Knowledge Organization System Reference*, Recommendation, World Wide Web Consortium (W3C):
<http://www.w3.org/TR/2009/PR-skos-reference-20090615/>; Lastest version available:
<http://www.w3.org/TR/skos-reference>
- Manola, F. dan Miller, E., 2004, *RDF Primer*, Recommendation, World Wide Web Consortium(W3C):
<http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>; Lastest version available:
<http://www.w3.org/TR/rdf-primer>
- Miles, A. dan Brickley, D., 2005, *SKOS Core Guide*, Recommendation, World Wide Web Consortium (W3C):
<http://www.w3.org/TR/2005/WD-swbp-skos-coreguide-20051102/>; available:
<http://www.w3.org/TR/swbp-skos-core-guide>
- Miles, A., 2005, *Quick Guide to Publishing a Thesaurus on the Semantic Web*, Recommendation, World Wide Web Consortium (W3C):
<http://www.w3.org/TR/2005/WD-swbp-thesaurus-pubguide-20050517/>; lastest version available:
<http://www.w3.org/TR/swbp-thesaurus-pubguide>
- Miles, A., Matthews, B., Beckett, D., Brickley, D., Wilson, M dan Rogers, N., 2005, *SKOS - A Language to describe simple knowledge structures for the web*, Recommendation, World Wide Web Consortium (W3C):
<http://www.w3.org/TR/swbp-skos:a language to describe simple knowledge structures for the web>